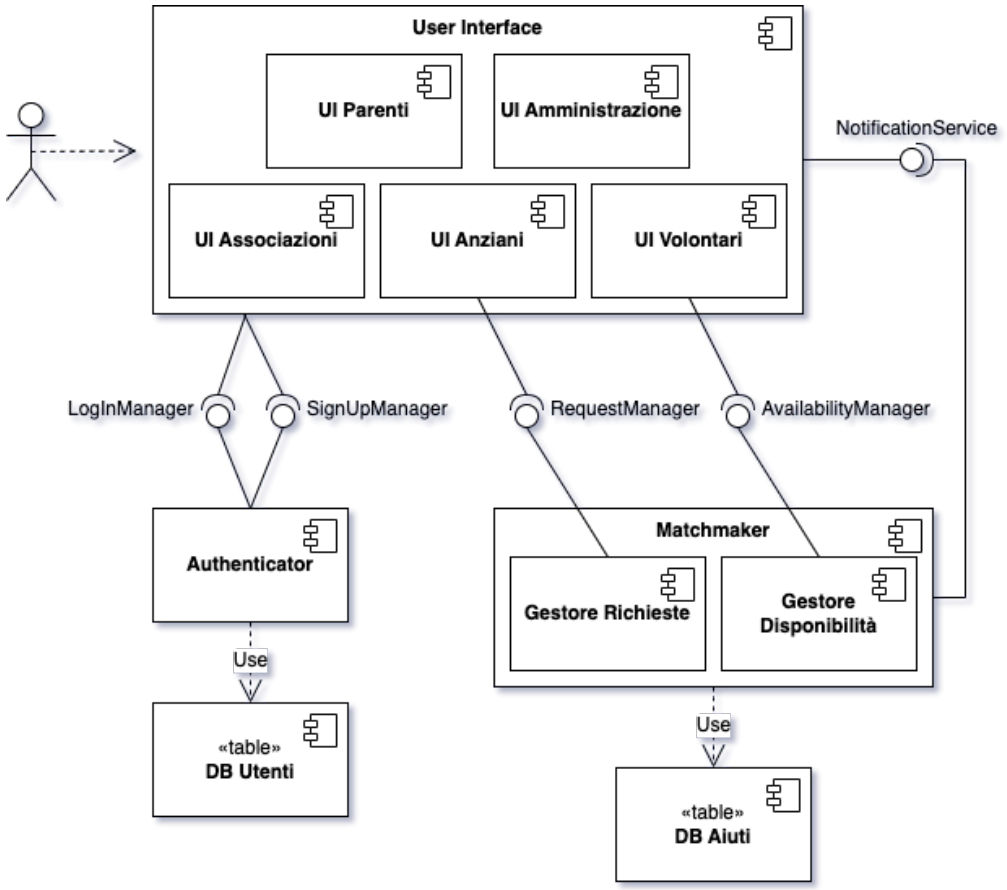
D2 - *Gener**Azioni***

Gruppo: **#8**
Membri: Tommaso Mario Repele **227120**, Tommaso Tamagnini **221448**

Diagramma dei componenti



User Interface

Il componente ***User Interface (UI)*** è la facciata dell'applicazione per tutti i tipi di utente, permette di accedere alle varie funzionalità offerte.

Nome	Tipologia	Descrizione
NotificationService	Fornita	Permette di mostrare la notifica di un evento all'utente (avvenuto match, timer match scaduto, nuova segnalazione, ecc).
LogInManager	Richiesta	Interfaccia dell'*Authenticator* che permette di riconoscere e fare il login degli utenti.

Nome	Tipologia	Descrizione
SignUpManager	Richiesta	Interfaccia dell'*Authenticator* che permette di registrare un nuovo utente.
RequestManager	Richiesta	Interfaccia del *GestoreRichieste* che permette di gestire le richieste di aiuto degli anziani (aggiungere, modificare, cancallere).
AvailabilityManager	Richiesta	Interfaccia del *GestoreDisponibilità* che permette di gestire le disponibilità dei volontari.

Authenticator

Il componente ***Authenticator*** si occupa dell'autenticazione degli utenti (login e registrazioni), permettendo di accedere al database degli utenti.

Nome	Tipologia	Descrizione
LogInManager	Fornita	Permette di verificare la correttezza delle credenziali fornite, quindi autenticare l'utente e sbloccare certe funzionalità.
SignUnManager	Fornita	Permette a un nuovo utente di immettere i propri dati e registrarsi nel sistema.
DBAccess	Richiesta	Interfaccia del *DBUtenti* che da' accesso in lettura e scrittura al database.

DBUtenti

Il componente ***DBUtenti*** rappresenta il database che contiene i dati degli utenti del sistema, sia le generalità che le credenziali di accesso.

Nome	Tipologia	Descrizione
DBAccess	Fornita	Permette di leggere e manipolare i dati del database.

Matchmaker

Il componente ***Matchmaker*** si occupa di trovare le corrispondenze tra richieste di aiuto delle persone anziane e disponibilità dei volontari. I sottocomponenti ***GestoreRichieste*** e ***GestoreDisponibilità*** si occupano di gestire richieste e offerte di aiuto.

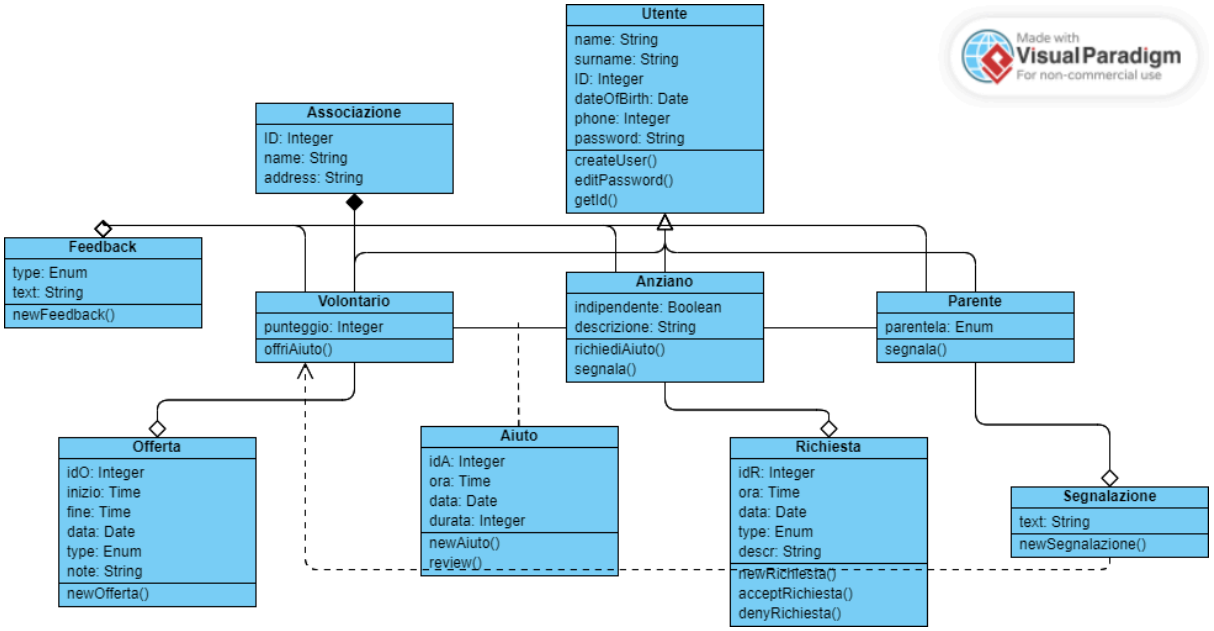
Nome	Tipologia	Descrizione
RequestManager	Fornita	Permette alle persone anziane di gestire le richiste d'aiuto.
AvailabilityManager	Fornita	Permette ai volontari di gestire le proprie disponibilità.
DBAccess	Richiesta	Interfaccia del *DBAiuti* che da' accesso in lettura e scrittura al database.
NotificationService	Richiesta	Interfaccia della *UI* che permette la segnalazione di un evento all'utente.

DBAiuti

Il componente DBAiuti rappresenta il database che contiene le richieste d'aiuto delle persone anziane e le disponibilità dei volontari; registra anche le sessioni d'aiuto completate, con i relativi feedback.

Nome	Tipologia	Descrizione
DBAccess	Fornita	Permette di leggere e manipolare i dati del database.

Diagramma e descrizione delle classi



Classi Utente

Questa classe comprende le sottoclassi Volontario, Anziano e Parente. I suoi attributi sono i dati relativi all'utente.

Operazione	**Descrizione**
createUser()	Permette la registrazione di un nuovo utente.
editPassword()	Permette di modificare la password di un account esistente.
getId()	Restituisce l'ID utente.

Sottoclasse Volontario

offriAiuto()	Permette di dare la propria disponibilità in un determinato giorno/fascia oraria per un determinato tipo di aiuto (crea una nuova Offerta).

Sottoclasse Anziano

richiediAiuto()	Permette di richiedere l'aiuto da parte di un volontario, specificando il tipo di aiuto

	richiesto, il giorno e l'ora (crea una nuova Richiesta).
segnala()	Permette di segnalare all'associazione una truffa subita o altri problemi con il volontario.

Sottoclasse Parente

segnala()	Analogo al metodo segnala() per la sottoclasse Anziano, solo che in questo caso la segnalazione avviene da parte di un volontario o persona di fiducia dell'anziano.

Classe Associazione

Aggregazione di classi Volontario. I suoi attributi sono l'ID associato, il nome e l'indirizzo dell'associazione.

Classe Offerta

Si occupa delle offerte di aiuto da parte dei volontari. I suoi attributi sono l'ID dell'offerta (per facilitarne la gestione) la fascia oraria (ora d'inizio e ora di fine) e la data in cui è disponibile il volontario, il tipo di aiuto richiesto ed eventuali annotazioni.

Operazione	**Descrizione**
newOfferta()	Crea una nuova offerta.

Classe Richiesta

Si occupa delle richieste di aiuto da parte degli anziani. I suoi attributi sono l'ID della richiesta (in modo da facilitarne la gestione), l'orario e la data in cui si richiede l'aiuto, il tipo di aiuto richiesto ed eventuali annotazioni.

Operazione	**Descrizione**
newRichiesta()	Crea una nuova richiesta.
acceptRichiesta()	Permette ad un volontario di accettare la richiesta e inizializza un Aiuto.
denyRichiesta()	Permette ad un volontario di non accettare la richiesta e inoltra la richiesta ad altri volontari.

Classe Aiuto

Si occupa di gestire le richieste di aiuto andate a buon fine (un Volontario aiuta un Anziano). I suoi attributi sono l'ID dell'aiuto (per facilitarne la gestione), l'ora e la data in cui un Volontario ha aiutato un Anziano e la durata dell'aiuto.

Operazione	**Descrizione**
newAiuto()	Inizializza un nuovo Aiuto quando un volontario accetta una richiesta di aiuto da parte di un anziano.
review()	Permette di scrivere un feedback a seguito dell'aiuto.

Classe Feedback

Si occupa di gestire i feedback da parte degli anziani/parenti oppure da parte dei volontari per un Aiuto. I suoi attributi sono il tipo di feedback (positivo/negativo) e il testo (descrizione).

Operazione	**Descrizione**
newFeedback()	Inizializza un nuovo Feedback.

Classe Segnalazione

Si occupa di gestire le segnalazioni effettuate dagli anziani o loro parenti/persone di fiducia di truffe ai danni degli anziani.

Operazione	**Descrizione**
newSegnalazione()	Inizializza una nuova Segnalazione.

Constraints OCL

Classi Utente

- Tutti gli utenti devono essere maggiorenni:
  ```
  context Utente
  inv: age >= 18
  ```
- Il numero di telefono inserito quando ci si registra deve essere valido:
  ```
  context Utente::createUser(name, surname, birth, phone, pass)
  pre: phone != NULL AND valid(phone)
  ```
- Le credenziali non possono essere vuote quando ci si registra:
  ```
  context Utente::createUser(name, surname, birth, phone, pass)
  pre: name != NULL AND surname != NULL AND birth != NULL AND
  phone != null AND pass != NULL
  ```
- La password deve contenere almeno 8 caratteri e meno di 32
  ```
  context Utente::createUser(name, surname, birth, phone, pass)
  pre: password.size()>=8 AND password.size()<=32
  ```
- Si può ottenere l'ID solo di un oggetto istanziato:
  ```
  context Utente::getId()
  pre: ID != NULL
  ```
- `editPassword()` modifica la password:
  ```
  context Utente::editPassword(pass)
  post: password = pass
  ```

Classe Richiesta

- `newRichiesta()` aggiunge una richiesta al database:
  ```
  context Richiesta::newRichiesta(ora, data, tipo, descr)
  post: DBAiuti.richieste.add(ora, data, tipo, descr)
  ```
- Una richiesta di aiuto non può essere fatta per un giorno nel passato:
  ```
  context Richiesta::newRichiesta(ora, data, tipo, descr)
  pre: data >= CurrentDate() AND ora > CurrentTime()
  ```

Classe Offerta

- `newOfferta()` aggiunge un'offerta al database:
  ```
  context Offerta::newOfferta(inizio, fine, data, tipo, descr)
  post: DBAiuti.offerte.add(inizio, fine, data, tipo, descr)
  ```
- Una offerta d'aiuto non può essere fatta per un giorno nel passato:
  ```
  context Offerta::newOfferta(inizio, fine, data, tipo, descr)
  pre: data >= CurrentDate() AND inizio > CurrentTime()
  ```
- La fine di una richiesta d'aiuto non può precedere l'inizio:
  ```
  context Offerta::newOfferta(inizio, fine, data, tipo, descr)
  pre: inizio < fine
  ```

Classe Aiuto

- `newAiuto()` aggiunge un aiuto al database:
  ```
  context Aiuto::newAiuto(ora, data, durata)
  post: DBAiuti.aiuti.add(ora, data, durata)
  ```
- Un aiuto non può essere fatto per un giorno nel passato:
  ```
  context Aiuto::newAiuto(inizio, fine, data, tipo, descr)
  pre: data >= CurrentDate() AND ora > CurrentTime()
  ```